George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

QD-QE-009
REVISION C
EFFECTIVE DATE: September 24, 2004

ORGANIZATIONAL INSTRUCTION

Software Assurance Review/Approval of Technical Documents

OPR(s)	**OPR DESIGNEE**
QD10,QD20,QD30,QD40	**Rosalynne Strickland**

DOCUMENT HISTORY LOG

Status (Baseline/ Revision/ Canceled)	Document Revision	Effective Date	Description
Baseline		10/13/00	
Revision	A	9/09/02	Format and numbering change to implement requirements of QD-A-001 rev F.
Revision	B	09/18/03	Changes made to incorporate new QD40 organizational name
Revision	C	9/24/04	Revised to bring document in compliance with the HQ Rules Review Action (CAITS: 04-DA01-0387). Changes were also made to reflect S&MA organizational name changes (i.e., QS to QD).

Software Assurance (SA) Review/Approval of Technical Documents

1.0 PURPOSE, SCOPE AND APPLICABILITY

1.1 Purpose - This organizational instruction provides the Software Assurance representative with direction for the review/approval of technical documents (and changes therein).

1.2 Scope - This organizational instruction applies to technical documents requiring review/approval (and changes therein) and includes, but is not limited to, program/project requirements documents, software requirements specifications, plans, and test procedures.

1.3 Applicability - This organizational instruction applies to Software Assurance by QD40.

2.0 DOCUMENTS (Applicable and/or Reference)

2.1 Applicable Documents

QD-QE-008 Software Assurance Status Report

2.2 Reference Documents

QD-QE-010 Software Assurance Software Milestone Review Support

QD-QE-011 Software Quality Assurance Software Configuration Audits

3.0 DEFINITIONS

3.1 Acceptance Testing - Testing conducted to determine whether or not a system satisfies its requirements and to determine whether or not to accept the system.

3.2 Baseline - A product that has been formally reviewed and approved that can be changed only through formal change control procedures.

3.3 Change Control - The process, by which a change to a baseline is proposed, evaluated, approved or rejected, scheduled, and tracked.

3.4 Component - A basic useable part of a system or program, an assembly of modules.

3.5 Configuration - A set of software, documentation, and data elements that meets a set of requirements or contractual obligations. The items that make up a baseline.

3.6 Configuration Control - Configuration control is a process to provide the administrative mechanism for precipitating, preparing, evaluation, and approving or disapproving all change proposals throughout the system life cycle. That is, software configuration control is change proposal processing.

3.7 Configuration Identification - Configuration identification includes the specifications and their associated diagrams, flow charts, drawings, parts lists, etc., that are used to describe the functional and physical characteristics of a CI.

3.8 Configuration Status Accounting and Reporting - Software configuration status accounting is the administrative tracking and reporting of all software items formally identified and controlled.

3.9 Computer Software Configuration Item - A defined software product that satisfies an end use function and is designated for configuration management.

3.10 Data Requirement Description/Data Requirements List (DRD/DRL) - DRD/DRL, or equivalent, are contract requirements, which specify the format, content, and delivery schedules for documentation required to be provided on the contract.

3.11 Design - The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system.

3.12 Firmware - Computer programs and data loaded in a type of memory that cannot be dynamically modified (i.e. PROMS, EPROM's).

3.13 Inspection - A formal technique in which SW requirements, design, or code are reviewed in detail by a person or group other than the author to detect faults, violations of standards, and other problems.

3.14 Module - A part of a computer program that is separable and identifiable with respect to compiling, combining with other parts, and loading. A subroutine is an example of a module.

3.15 Nonconformance - Any deviation of hardware, software, or documentation from its functional, performance, or interface requirements or from the standards to which it is to be developed.

3.16 Regression Testing - Re-testing done after program modification to verify that the modifications have not introduced faults or unintended adverse side effects.

3.17 Review Item Discrepancy (RID) - A form used to document errors, inconsistencies, or omissions discovered in a document, drawing, or procedure being evaluated for acceptability during a formal project review.

3.18 Security - The protection of computer hardware and software from accidental and deliberate unauthorized access, use, modification, destruction, or disclosure.

3.19 Software - Computer programs, procedures, associated documentation and data. At MSFC the term software includes firmware.

3.20 Software Assurance -The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures. Software Assurance includes the disciplines of Software Quality (functions of Software Quality Engineering, Software Quality Assurance, Software Quality Control), Software Safety, Software Reliability, Software Verification and Validation, and Independent Verification and Validation (IV&V).

3.21 Computer Software Configuration Item - A defined software product that satisfies an end use function and is designated for configuration management.

3.22 Software Configuration Management (SCM) - A discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of software configuration items and baseline, (2) control changes to those characteristics, and (3) record and report change processing and implementation status of software configuration items and baseline.

3.23 Software Development Agent (SDA) - The NASA organization or contractor responsible for software management, development, and assurance. For in-house software development, the contract may be the Project Plan, Software Development Plan, or other governing document.

3.24 Software Quality Assurance (SQA) - The planned, systematic process that ensures that desired procedures, standards, requirements, and quality attributes are:

- Established prior to software acquisition/development
- Followed during each phase of acquisition/development

Ultimately, the basic Software Quality Assurance function is to ensure that both software products and acquisition process comply with established standards, practices, and procedures.

3.25 Software Library - A collection of software and related documentation that is designed to aid in software development, use, maintenance, or control.

3.26 System - A collection of software programs organized to accomplish a specific set of functions or to meet a set of requirements.

3.27 Test Plan - A document describing the approach to a planned testing activity. The plan typically identifies the requirements and items to be tested, the test to be run, test schedules, resources and data requirements, reports to be produced, and evaluation criteria.

3.28 Test Procedure - A document giving detailed instructions for the setup, operation, and results for a given test.

3.29 Tool - A program used to assist in the development, testing, analysis, or maintenance of another computer program or its documentation.

3.30 Unit - Separately named and accessible elements of software, which perform specific functions. Also called subroutines, functions, procedures, or modules.

3.31 Unit Development Folder - A formalized set of records documenting the development of a software unit. It includes schedules, reviews, approvals, and supporting documentation.

3.32 Validation - The process of evaluation of software to assure that it meets its requirements. It is normally done by reviews and testing.

3.33 Verification - The process of determining whether the products of a given phase of the software development cycle fulfill the requirements established during the previous phase

4.0 INSTRUCTIONS

4.1 Technical Document Review - The review of technical documents by the Software Assurance representative shall assure the following:

4.1.1 All documents contain adequate criteria for acceptance of associated software and processes, and for test performance.

4.1.2 All requirements and criteria are clear and concise and meet project/program requirements.

4.1.3 Documents contain sufficient information to initiate preliminary software quality planning functions.

4.1.4 Test procedures contain adequate requirements to comply with test specifications.

4.2 Software Development Plan (SDP) - The Software Development Plan (SDP) documents the software developing organization's understanding of the project/program's software requirements, and indicates how it will manage and control the software development activities. The Software Assurance representative may be requested to support the project/program by reviewing and evaluating this SDP. The following instructions address the procedures used in reviewing the SDP:

4.2.1 Review the contract Statement of Work (SOW) and DRD/DRL to determine the management controls, design reviews, testing and monitoring requirements.

4.2.2 Tailor the checklist (see Appendix A) to reflect the SOW and DRD/DRL requirements.

4.2.3 Evaluate the document using tailored checklist.

4.2.4 Document the results of the SDP evaluation in a SA Status Report, QD-QE-008.

4.3 Supplier-Prepared Software Quality Assurance (SQA) Documentation - The requirements for a supplier's SQA program is documented in the contract SOW and the DRD/DRL. Prior to implementing the plans, the supplier shall submit their plan to the Safety and Mission Assurance Office for review/approval, when required.

4.3.1 SQA Plan - The objective of the review is to verify the supplier's SQA plan is fully integrated with the development process and satisfies the SQA requirements. The following instructions address the procedures used in reviewing the supplier's SQA plan:

4.3.1.1 Review the contract SOW and DRD/DRL for the SQA activities.

4.3.1.2 Tailor the checklist (see Appendix B) to ensure compatibility with the contract.

4.3.1.3 Evaluate the SQA Plan using the tailored checklist.

4.3.1.4 Document the results of the SQA Plan evaluation in a Software Assurance Status Report, QD-QE-008.

4.3.2 SQA Procedures - The objective of the review is to verify the supplier's SQA procedures satisfies the SQA requirements. The following instructions address the procedures used in reviewing the supplier's SQA procedures:

4.3.2.1 Tailor the checklist (see Appendix C) to ensure compliance with the SQA Plan.

4.3.2.2 Evaluate the SQA procedures using the tailored checklist.

4.3.2.3 Document the results of the SQA procedures evaluation in a Software Assurance Status Report, QD-QE-008.

4.4 Software Requirements Specification - The objective of an SRS is to establish the software requirement for the subsequent development. The following instructions address the procedures used in reviewing the SRS:

4.4.1 Review the appropriate higher-level requirements documents (i.e. SOW, DRD/DRL, System Documents, CEI, etc.)

4.4.2 Tailor the checklist (see Appendix D to ensure compatibility with the contract.

4.4.3 Evaluate the SRS using the tailored checklist.

4.4.4 Document the results of the SRS evaluation in a SA Status Report, QD-QE-008.

4.5 Test Documentation - The test organization is responsible for developing, preparing, distributing, and obtaining approval of test documentation. This documentation shall contain the minimum requirements necessary to assure product compliance with design requirements and specifications.

4.5.1 Test Plan - The objective of the review is to verify the test plan satisfies the testing requirements. The following instructions address the procedures used in revising the test plan:

4.5.1.1 Review the contract SOW and DRD/DRL to determine the testing requirements.

4.5.1.2 Tailor the checklist (see Appendix E) to reflect the SOW and DRD/DRL requirements.

4.5.1.3 Evaluate the test plan using the tailored checklist.

4.5.1.4 Document the results of the test plan evaluation in a Software Assurance Status Report, QD-QE-008.

4.5.2 Test Procedures - The objective of the test procedure review is to verify the test procedures satisfies the testing requirements. The following instructions address the procedures used in evaluating the test procedures:

4.5.2.1 Tailor the checklist (see Appendix F) to ensure compliance with the test plan.

4.5.2.2 Evaluate the test procedures using the tailored checklist.

4.5.2.3 Document the results of the test procedures evaluation in a Software Assurance Status Report, QD-QE-008.

4.5.3 Test Report - The objective of the test report review is to verify the test report satisfies the test requirements. The following instructions address the procedures used in evaluating the test report:

4.5.3.1 Review the SOW and DRD/DRL to determine the testing requirements.

4.5.3.2 Tailor the checklist (see Appendix G) to reflect the SOW and DRD/DRL requirements.

4.5.3.3 Evaluate the test plan using the tailored checklist.

4.5.3.4 Document the results of the test plan evaluation in a Software Assurance Status Report, QD-QE-008.

4.6 Other Documentation - The objective for reviews of documents not listed will be to ensure they satisfy the applicable requirements. The following instructions address the procedures used in evaluating "other" documentation:

4.6.1 Review the contract documents to ensure compliance.

4.6.2 Generate checklist to reflect contract requirements documents.

4.6.3 Evaluate the document using the checklist generated.

4.6.4 Document the results of the document evaluation in a Software Assurance Status Report, QD-QE-008.

4.7 Approval of Technical Documentation - The Software Assurance representative shall have approval/concurrence authority for technical documents.

5.0 NOTES

None

6.0 SAFETY PRECAUTIONS AND WARNING NOTES

None

7.0 APPENDICES, DATA, REPORTS, AND FORMS

APPENDIX A – Software Development Plan Evaluation Checklist

APPENDIX B – Software Quality Assurance Plan Evaluation Checklist

APPENDIX C – Software Quality Assurance Procedures Evaluation Checklist

APPENDIX D – Software Requirements Specification Evaluation Checklist

APPENDIX E – Software Test Plan Evaluation Checklist

APPENDIX F – Software Test Procedure Evaluation Checklist

APPENDIX G – Software Test Report Evaluation Checklist

8.0 RECORDS

None

9.0 TOOLS, EQUIPMENT, AND MATERIALS

None

10.0 PERSONNEL TRAINING AND CERTIFICATION

None

11.0 FLOW DIAGRAM

None

12.0 RESPONSIBILITIES

Work accomplished within the scope of this organizational instruction will performed by the Software Assurance representative. The Safety, Reliability, and Quality Assurance Policy and Assessment Department (QD40) may delegate the responsibilities and tasks provided in this organizational instruction to support contractors who are responsible for carrying out the tasks identified herein.

APPENDIX A – Software Development Plan Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	The organization of the software design and development work force is depicted clearly.		
2.	The roles, missions, and authorities of support organizations are clearly described.		
3.	The master schedule is presented in detailed.		
4.	Policies and Procedures for ensuring management visibility and control are given.		
5.	The policies and procedures for ensuring positive, effective control of performance and design are given.		
6.	The policies and procedures for ensuring quality, both in regard to software development methodology and deliverable products, are given.		
7.	The policies and procedures for effecting positive control over, and securing an audit trail of, work-in-process are given. The methodology for managing the design change is clear.		
8.	The sequence of activities that make up the development process is described. Interdependencies are discussed.		
9.	Programming standards, conventions, and design rules have been defined for all levels of implementation.		
10.	Guidelines are given for establishing, monitoring, and adjusting software timing and sizing budgets. A specific reference is made, and methods described, for dealing with those programs that are expected to be time-critical and resource-critical.		
11.	A procedure exists for reporting and resolving software errors.		
12.	All non-deliverable software is specified and need dates are given.		
13.	Equipment and facility requirements are given and the planned schedule for satisfying them is evident.		
14.	Staff's training requirements are given and the planned schedule for satisfying them is evident.		
15.	Test philosophy is described and test management/control procedures are identified.		
16.	Risk Management procedures are described.		
17.	Procedures for supporting formal reviews are defined.		
18.	Procedures for conducting testing are defined.		

APPENDIX B – Software Quality Assurance Plan Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	Is the SQA plan in compliance with the DRD/DRL requirements?		
2.	Are all contractual requirements addressed in the plan?		
3.	Are the duties and responsibilities of the SQA organization defined?		
4.	Does the SQA organization have an independent reporting channel to appropriate levels of management so that quality problems and conflicts can be efficiently and effectively resolved?		
5.	Has the SQA plan establish a training, evaluation, and certification program for SQA personnel?		
6.	Are the life cycle audits, reviews, and inspections adequate to ensure compliance with requirements, processes, procedures, and standards?		
7.	Does the record retention system provide an adequate audit trail?		
8.	Are there plans for conducting audits on the Configuration Management (CM) practices?		
9.	Are there plans for conducting audits on the procedures related to the development and delivery of software documentation?		
10.	Does the supplier maintain a documented system for the handling of nonconformance reports?		
11.	Does the supplier maintain a system for taking corrective action in order to prevent repetitive nonconformances?		
12.	Is the supplier's corrective action system one that permits prompt and remedial action?		
13.	Does the supplier maintain a system for following up on all corrective action requests?		
14.	Are the reports on nonconforming practices regularly prepared and reviewed by management for action and status?		
15.	Does the SQA plan apply the SQA process to COTS and GFE software and firmware?		
16.	Do SQA personnel participate in the test process?		
17.	Do SQA personnel participate in the design review and baseline process?		
18.	Do SQA personnel participate in the software delivery and acceptance process?		

APPENDIX B (continued)– Software Quality Assurance Plan Evaluation Checklist

19.	**Does the SQA process include requirements for conducting internal audits on non-deliverable software?**		
20.	**Are the SQA requirements flowed down to subcontractors?**		
21.	**Does the prime contractor approve subcontractor's SQA plan?**		
22.	**Are there provisions for the Safety, Reliability, and Quality Assurance Department (QD10) SQA representative visibility into subcontractor's practices?**		
23.	**Does the SQA plan identify the prime contractor's monitoring and auditing of the subcontractor's SQA implementation?**		

APPENDIX C - Software Quality Assurance Procedures Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	Do the procedures address the deliverable and non-deliverable software and firmware?		
2.	Is there a software development plan (SDP) on this project/program?		
3.	Does SQA review and approve the SDP?		
4.	Is there a Software Configuration Management Plan (SCMP) on this project/program?		
5.	Does SQA review and approve the SCMP?		
6.	Is there an SQA audit schedule?		
7.	Are the project and software development organizations defined and documented?		
8.	Is the organizational independence and authority for execution of software quality evaluation documented?		
9.	Does SQA review all documentation prior to delivery?		
10.	Does SQA review and approve all documentation development and delivery plan?		
11.	Does SQA ensure that the software design standards in use address: a) Approved languages? b) Control structures? c) Module/Unit size? d) Branching? e) Symbolic parameters? f) Naming conventions?		
12.	Does SQA participate in development (informal) testing?		
13.	Does SQA participate in formal testing?		
14.	Does SQA verify that testing requirements are evaluated and documented?		
15.	Does SQA verify that tests are conducted in accordance with approved procedures?		
16.	Does SQA verify documentation of test failures?		
17.	Does SQA verify all computer software products and associated documentation are in a controlled status?		
18.	Does SQA verify differences between expected and observed test results are reconciled?		

APPENDIX C (continued) – Software Quality Assurance Procedures Evaluation Checklist

19.	**Does the project/program have a system established and documented for detecting, reporting, and correcting software problems?**		
20.	**Does SQA report, document, and ensure closure of all discrepancy/action item assignments made to the project/program?**		
21.	**Has a system of SQA trend analysis been established and documented?**		
22.	**Does the trend analysis system provide for the use of trend data as a basis for management actions?**		
23.	**Does SQA have a documented plan for conduct of SQA audits and reviews?**		
24.	**Does the SQA review and audit plan cover the following areas:** **a) Project/Program design review process?** **b) Project Management controls?** **c) DRDs/DRLs?** **d) Software Configuration Management?** **e) Testing?** **f) Library Controls?** **g) Programming Standards?** **h) Audit of the SQA organization?**		
25.	**Does the SQA procedures provide for establishment of records of all SQA audits, inspections, and tests?**		

APPENDIX D – Software Requirements Specification Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	**Have all issues with higher-level requirements been resolved?**		
2.	**Were the higher-level requirements baselined prior to initiating the software requirements definition?**		
3.	**What changes have been approved to the higher-level requirements since beginning of the software requirements definition process?**		
4.	**Have the approved changes been formally communicated to the impacted organizations?**		
5.	**Have the higher-level requirements been updated to reflect approved changes?**		
6.	**Are the requirements being detailed in a consistent manner?**		
7.	**Is there traceability between higher-level requirements document and the SRS?**		
8.	**Are the software interfaces identified and defined?**		
9.	**Are there functional flow diagrams (or equivalent) showing the software functions and interrelationships?**		
10.	**In determining the techniques to be used in displaying information, has consideration been given to the following:** **a) Formats to be used when presenting information?** **b) Coding conventions to be employed in presenting information details?** **c) Rules for routing displays?** **d) Rules for forcing displays?** **e) Rules for updating displays?** **f) Rules for priority handling of displays?**		
11.	**Have all approved changes impacting the software functions, performance, and data been accounted for in the detailed requirements definition?**		
12.	**Have design feasibility studies been conducted in support of the requirements definition?**		
13.	**Has there been a risk assessment conducted and documented?**		
14.	**Are the database requirements defined?**		
15.	**Have the software performance requirements been defined?**		

APPENDIX D (continued)– Software Requirements Specification Evaluation Checklist

16.	**Have the requirements for each operational workstation been defined in terms of the following:** **a) Information required?** **b) Required evaluation, decisions, and coordination?** **c) Required actions?** **d) Frequencies of actions?** **e) Available controls?** **f) Nature of feedback?**		
17.	**Have the higher-level software quality requirements been detailed and included in the SRS?**		
18.	**Have standards been considered and identified for use during software development and implementation?**		

APPENDIX E – Software Test Plan Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	Tailor checklist to specific needs.		
2.	Requirements for implementing and controlling the test environment resources must be addressed. The test resources should be include in the identification of the following: a) Software items (e.g. test drivers, compilers, operating systems, etc.) necessary to perform the formal test activities. b) Hardware and firmware items (computer hardware, interfacing equipment, firmware items) that will be used in the software test environment.		
3.	Developer's plan for installing, pre-testing, and controlling/maintaining the support software should be described.		
4.	Plan must identify responsibilities of the various organizations involved in the testing. This should include: a) Test conduct (responsible organization) b) Test witnessing (who and how)		
5.	There should be a plan for generating the test procedures that address who prepares, reviews, and approves, format, content and control of procedure.		
6.	The plan should address support documentation and procedures such as test preparation sheets, nonconformance reporting, failure reporting and test reports.		
7.	There must be a requirement for conducting a pretest review (i.e. TRR) and posttest meeting with the test team.		
8.	The location and schedule for the test conduct must be identified.		

APPENDIX F – Software Test Procedure Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	Tailor checklist to specific needs of the design requirements and specifications.		
2.	The procedure must have the location and schedule for the following activities: a) Briefings b) Pretest activities c) Test conduct d) Debriefings e) Data reduction and analysis		
3.	There should be pretest procedures for preparation and set up of the test environment.		
4.	There should be step-by-step instructions for loading the software being tested.		
5.	There should be step-by-step instructions for loading the support software.		
6.	Procedures should identify when the support software is loaded.		
7.	There should be a test description for the test to be conducted.		
8.	Traceability should be between the test cases and the SRS.		
9.	Procedures should be specify the prerequisite conditions that must be established prior to performing the test cases.		
10.	All test inputs should be specified and for each input the following data should be provided: a) Name, purpose, and description b) Source of test input and method to be used for selecting it. c) Whether the input is real or simulated. d) Time or event sequence of the test input.		
11.	The expected results should be specified in terms of intermediate and final results.		
12.	The criteria for evaluating the final results should be specified.		
13.	The procedures for conducting the test should be defined. Test operators actions, expected results, any actions in the event of error, and process used in data reduction.		

APPENDIX G – Software Test Report Evaluation Checklist

Item	Criteria	Assessment	Comments
1.	**Tailor checklist to specific needs.**		
2.	**A summary of the test results should be given.**		
3.	**A chronological record of all events relevant to test preparation, test performance, analysis and interpretation of formal test results should be given.**		
4.	**For each step of the test procedure executed there should be a result recorded.**		
5.	**Deviations from the test procedure must be identified.**		
6.	**The analysis should identify any deficiencies, limitations, or constraints in the product that were detected by the test performed.**		
7.	**Problem/change reports shall be generated for each deficiency identified; and for each deficiency, limitation, or constraint, the analysis should:** **a) Describe its impact on product and system performance.** **b) Describe the impact on the product and system design in order to correct it.** **c) Provide a recommended solution/approach for correcting it.**		